National University of Singapore
School of Computing

IS2103 Enterprise Systems Development Concepts

Part 2: Enterprise JavaBeans
MRS: Module Registration System

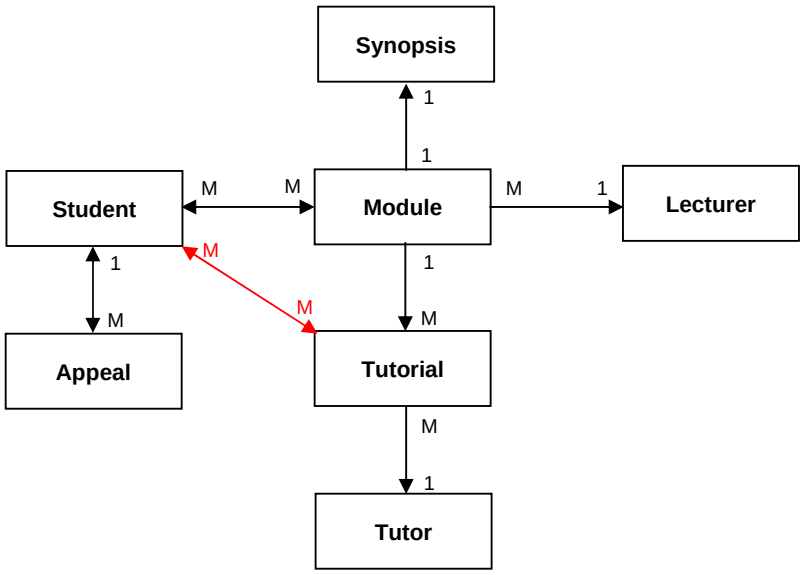
Total Marks: 30

DOMAIN INFORMATION

Department of Information Systems is going to launch an online system to facilitate module and tutorial registration.

Students can register modules and tutorials via an online system. Every student has to login the Module Registration System (MRS) before he/she can use the system functionalities. In addition, students can use MRS to search for modules, view the details of the module and tutorial information, and register modules and tutorials. MRS also provides other functionalities to facilitate module and tutorial management.

From a detailed study of the module registration processes at the department, an entity relationship diagram is provided:



Each student is given an account with a default password. A student can change the password and personal information after login. He/she can search for modules using either module code or keywords of module title, and view the detail information of each module, including lecturer, time, venue, synopsis and corresponding tutorials (if any). He/she can also register modules and tutorials on the system, and view his/her timetable.

A system administrator is able to manage student accounts, modules and tutorials.

The department also maintains a system to manage module/tutorial registration appeals from students. An appeal is sent from a non-web-based client. System administrator will view and process the appeals accordingly.

Each entity in the entity relationship diagram has a set of attributes (i.e. properties or data fields) that describes its static information. The list below defines the attributes. *Note that the list does not include data fields for managing entity relationships. The set of attributes is sufficient for this assignment.*

Student
- Id (unique identifier)
- Matric Number
- Name
- Password[1]
- Email

Module
- Id (unique identifier)
- Module Code
- Module Title
- Time
- Venue

Lecturer
- Id (unique identifier)
- Staff Number
- Title
- Name
- Email
- Office

Synopsis
- Id (unique identifier)
- Content

Tutorial
- Id (unique identifier)
- Group Number
- Time
- Venue

Tutor
- Id (unique identifier)
- Staff Number
- Name
- Email

Appeal
- Id (unique identifier)
- Time
- Content
- Status
- Comment

[1] There is no requirement for password to be encrypted in this assignment

SYSTEM REQUIREMENTS

Part 2a: Module Registration System (for the Administrator)

1. Given the above domain information, produce an EJB 3.0 application for the Module Registration System with a ***non-web-based client administration application***.
2. The application will be called "MRS: Module Registration System".
3. The entities given above are sufficient for your application. You DO NOT HAVE TO CREATE ADDITIONAL ENTITIES FOR THIS ASSIGNMENT.
4. The client application that you build will be used to test your EJB 3.0 application. For evaluation purpose, you are required to produce the following functionalities[2] in the client application[3]:
    a. Add student accounts. Allows an administrator to add student accounts into the system. If the account is already existing, an error message will be shown.
    b. Delete student accounts. Allows an administrator to remove student accounts from the system. If the student is not existing, an error message will be shown. A student account cannot be deleted if it is associated with a module or an appeal.
    c. Add lecturers. Allows an administrator to add lecturers into the system. If the lecturer is already existing, an error message will be shown.
    d. Delete lecturers. Allows an administrator to remove lecturers from the system. If the lecturer is not existing, an error message will be shown. A lecturer cannot be deleted if it is associated with a module.
    e. Add modules. Allows an administrator to add modules into the system. Each module must be associated with a lecture and a synopsis. If the module is already existing, an error message will be shown. For simplicity, we assume that the vacancy/quota for module is unlimited.
    f. Delete modules. Allows an administrator to remove modules from the system. If the module is not existing, an error message will be shown. The deleted module is disassociated from the lecturer, and its associated synopsis and tutorials are also deleted. A module cannot be deleted if it is registered by a student, or one of its tutorials is register by a student.
    g. Add tutors. Allows an administrator to add tutors into the system. If the tutor is already existing, an error message will be shown.
    h. Delete tutors. Allows an administrator to remove tutors from the system. If the tutor is not existing, an error message will be shown. A tutor cannot be deleted if it is associated with a tutorial.
    i. Add tutorials. Allow an administrator to add tutorials to a given module code. If the tutorial is already existing, an error message will be shown.
    j. Delete tutorials. Allows an administrator to remove tutorials from the system. If the tutorial is not existing, an error message will be shown. The deleted tutorial is disassociated from the tutor. A tutorial cannot be deleted if it is registered by a student. For simplicity, we assume that the vacancy/quota for each tutorial session is unlimited.
    k. View modules. Allows an administrator to view detail information of a given module code, including module title, lecturer, time, venue, synopsis, list of registered students (name, matric number and email).

[2] For part 2a and 2b, we will not be evaluating validation checks. That is, if an integer value is required for a data entry, we will enter the appropriate integer value.
[3] How the client application execution flows is left to your own design. Bonus marks will be given for good user interface design.

l. View tutorials. Allows an administrator to view detail information of all tutorials of a given module code, including the group number, time, venue, tutor and list of registered students (name, matric number and email) of each tutorial.
m. Assign students to modules. Allows an administrator to assign a student to a module, given the matric number, module code. If the student or the module is not existing, an error message will be shown. For simplicity, we assume that the administrator has been aware that there is no time clash when assigning a student to a module.
n. Assign students to tutorials. Allows an administrator to assign a student to a tutorial, given the matric number, module code, tutorial group number. If the module, the tutorial group or the student is not existing, an error message will be shown. For simplicity, we assume that the administrator has been aware that there is no time clash, and this student has not registered any tutorial of this module when assigning a student to a tutorial.
o. Process appeals. Allows an administrator to view the appeals sent from the students. The list of appeals are displayed on a first-come-first-served basis. The administrator can select an appeal by appeal Id and update the appeal status from "unread" to "approved", "rejected" or "processing". The administrator can also add some comments to the appeal. For simplicity, we assume that the administrator will manually update necessary information via other functionalities when handling appeals. i.e. You don't need to handle auto updating module/tutorial information when an appeal is processed.

Part 2b: Module Search System (for the Student)

1. Produce a ***Web-based client application using JSP and Servlet*** for the same Module Search System in Part 2a.
2. The entities given above are sufficient for your application. You DO NOT HAVE TO CREATE ADDITIONAL ENTITIES FOR THIS ASSIGNMENT.
3. The Web-based client application will be used concurrently with the non-web-based client application to test your EJB 3.0 application. For evaluation purpose, you are required to produce the following functionalities in the Web-based client application[4] specially designed for customer's use:
    a. Login the system. A student has to login using the matric number and password before he/she can use the following functionalities.
    b. Update profiles. A student can change the password and update his/her email on the system.
    c. Provide a search for modules. A student can search for module information by specifying the module code or keywords in the module title. A list of modules will be displayed[5] together with the list of tutorials of each module.
    d. Facilitate module registration. A student can register modules on the system. The system is able to check the student's timetable. No time clash is allowed and corresponding warning message should be displayed.
    e. Facilitate tutorial registration. A student can register tutorials for the module registered on the system. Similarly, the system is able to check timetable clash, and display warning message.

[4] How the web-based client application execution flows is left to your own design.
[5] Display module code, title, synopsis, lecturer, time, and venue. For tutorials, display group number, tutor, time and venue.

f. Facilitate the viewing of timetable. A student is able to view his/her timetable based on his/her module and tutorial registered on the system.
g. Facilitate the viewing of appeals. A student is able to view the content, status and comments of his/her appeals.

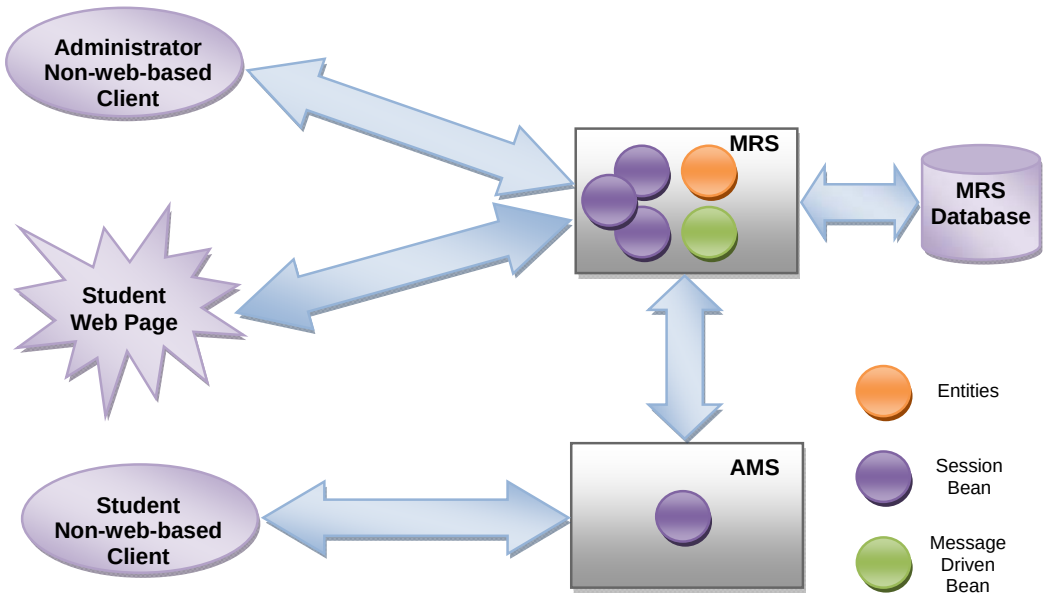
Implementation

1. In your implementation of the MRS and prior to the evaluation of your assignment, your system should have:
    a. At least 5 students in your implementation.
    b. At least 8 modules in your implementation.
    c. At least 2 tutorials per module in your implementation.

Part 2c: AMS: Appeal Management System

1. Produce another EJB 3.0 application on a **separate Netbeans project** to facilitate sending appeals.
2. The application will be called "AMS: Appeal Management System".
3. This application will have a ***non-web-based client*** to facilitate sending appeals**.** No login is required. A student only needs to input his/her matric number and appeal content. The appeal time is system-generated, and the appeal status is set to "unread" by default.
4. AMS has a session bean that receives the appeals from the students and sends a message to MRS. A message-driven bean in MRS receives the appeals and updates it into a database. No acknowledgement is required. The appeals can be viewed and processed by administrator in the administrator client application as described in Part 2a.

Pictorially, MRS and AMS communication looks like this:



EVALUATION

1. There will be an evaluator assigned to assess your work during the assignment evaluation week.
2. You will be required to deploy the EJB application on a computer in the laboratory or on your own laptop.
3. Tests on the functionalities of the non-web-based and web-based client applications will be carried out by the evaluator simultaneously. The evaluator will be examining the source code of the client and server logs of the EJB 3.0 application to ascertain the proper use of Enterprise JavaBeans 3.0 concepts.
4. Note that all database queries within your EJB application should be done using EJB-QL. ***Marks will be deducted for the use of native SQL in database queries.***
5. Marks[6] will be allocated in the following manner for all *executable* applications:
    a. MRS Functionalities (up to a maximum of 26 marks)
        - MRS for Administrator (up to 14 marks)
        - MRS for Student (up to 12 marks)
    b. AMS Functionalities (up to a maximum of 4 marks)
6. Bonus marks: based on additional features such as good and nice web page design, advanced search function, encrypted passwords, or any function that has not been specified above (up to a maximum of 3 bonus marks will be awarded).
7. If your EJB 3.0 application is unable to deploy and therefore not executable, up to 3 marks will be awarded for your Part 2 submission.
8. If your EJB 3.0 application is able to deploy and is executable, but none of the functionalities required in Part 2 works properly, up to a maximum of 5 marks will be awarded for your Part 2 submission.
9. Please be reminded that plagiarism is a serious offence. Disciplinary actions will be taken against those caught cheating.

[6] A student can score up to a MAXIMUM of 30 marks for Part 2 (including bonus marks).